

IN THE CLAIMS

Please cancel claim 13.

Please amend the claims as follows.

- 1 1) (Original) An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor; and
4 a database optimizer residing in the memory and executed by the at least one
5 processor, the database optimizer using statistics regarding the type of applications
6 accessing data in a database, the frequency with which the applications access the data,
7 and the location of the data being accessed by the applications to make at least one
8 change to the database schema to optimize the performance of accessing data in the
9 database.

- 1 2) (Original) The apparatus of claim 1 wherein the database optimizer makes the
2 change to the database schema according to a set of rules that specify a preferred data
3 type for each type of application accessing data in the database.

- 1 3) (Original) The apparatus of claim 1 wherein the change to the database schema
2 comprises changing the data type of at least one column in the database.

- 1 4) (Original) The apparatus of claim 1 wherein the change to the database schema
2 comprises adding a new column of a second data type to the database that contains the
3 same data in an existing column of a first data type in the database.

- 1 5) (Original) The apparatus of claim 4 wherein the database optimizer further
2 comprises a data coherency mechanism for maintaining data coherency between the
3 existing column and the new column.

1 6) (Original) The apparatus of claim 1 wherein the database optimizer receives
2 requests from at least one application to access data in the database, and returns data from
3 the database of a data type that is expected by the requesting application.

1 7) (Original) The apparatus of claim 1 wherein the database optimizer further
2 comprises a run-time statistics gathering mechanism to gather the statistics.

1 8) (Original) The apparatus of claim 1 wherein the database optimizer operates
2 according to customization settings set by a human user.

1 9) (Original) The apparatus of claim 1 wherein the database optimizer further
2 comprises a data type conversion mechanism that converts data in a first data type
3 retrieved from the database to a second data type that is preferred by an application
4 requesting the data.

1 10) (Original) An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database residing in the memory;
5 a database optimizer residing in the memory and executed by the at least one
6 processor, the database optimizer comprising:
7 a data access mechanism that uses statistics regarding the type of
8 applications accessing data in a database, the frequency with which the
9 applications access the data, and the location of the data being accessed by the
10 applications to make at least one change to the database schema to optimize the
11 performance of accessing data in the database;
12 customization settings that may be set by a human user to determine the
13 function of the database optimizer;
14 a data coherency mechanism that maintains coherency of reflective
15 columns in the database that are created by the data access mechanism and that
16 contain the same data in different data types; and
17 a data type conversion mechanism that converts data in a first data type
18 retrieved from the database to a second data type that is preferred by a requesting
19 application.

1 11) (Original) An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database residing in the memory;
5 a database optimizer residing in the memory and executed by the at least one
6 processor, the database optimizer comprising:
7 a mechanism that reads statistics regarding the type of applications
8 accessing data in the database, the frequency with which the applications access
9 the data, and the columns being accessed by the applications;
10 if the statistics indicate that a selected type of application has a number of
11 accesses to a selected column of a first data type in the database that exceeds a
12 first threshold level, the database optimizer determines whether the statistics
13 indicate that the selected type of application has a number of accesses to the
14 selected column that exceeds a second threshold level, and if so, the database
15 optimizer changes the data type of the selected column in the database;
16 if the statistics indicate that a selected type of application has a number of
17 accesses to a selected column of a first data type in the database that exceeds a
18 first threshold level, the data optimizer determines whether the statistics indicate
19 that the selected type of application has a number of accesses to the selected
20 column that exceeds a second threshold level, and if not, the data optimizer adds a
21 new column of a second data type to the database that contains the same data in
22 the selected column, the selected column and the new column being defined as
23 reflective columns because they contain the same data in different data types;
24 wherein the data optimizer detects when one of the plurality of
25 applications requests access to data in the selected column, determines the
26 preferred data type for the requesting application, determines if the data in the
27 selected column is of the preferred data type for the requesting application, and if

(claim 11 continued)

28 the data in the selected column is of the preferred data type for the requesting
29 application, returning the data in the selected column to the requesting
30 application;

31 if the data in any column reflective of the selected column is of the
32 preferred data type for the requesting application, the database optimizer returns
33 the data from the reflective column to the requesting application;

34 if the data in the selected column and in all reflective columns, if any, is
35 not of the preferred data type for the requesting application, the database
36 optimizer converts the data to the preferred data type for the requesting
37 application, and returns the converted data to the requesting application.

1 12) (Currently amended) A method for optimizing a database comprising the steps of:
2 (1) determining a preferred data type for at least one of a plurality of applications
3 that access the database; [and]
4 (2) dynamically changing a schema for the database to provide the preferred data
5 type when at least one of the plurality of applications requests access to data in the
6 database; and
7 wherein the step of dynamically changing the schema determines what change to
8 make according to:
9 2A) the type of the plurality of applications accessing data in the database;
10 2B) the frequency with which the plurality of applications access the data;
11 and
12 2C) the location of the data being accessed by the plurality of applications.

1 13) (Cancelled)

1 14) (Original) The method of claim 12 further comprising the steps of:
2 (3) determining when one of the plurality of applications accesses the database
3 that has a different preferred data type than the data type specified in the database
4 schema; and
5 (4) converting the data retrieved from the database to the different preferred data
6 type.

1 15) (Original) The method of claim 12 wherein step of dynamically changing the
2 schema for the database comprises the step of changing the data type of at least one
3 column in the database.

1 16) (Original) The method of claim 12 wherein the step of dynamically changing the
2 schema for the database comprises the step of adding a new column of a second data type
3 to the database that contains the same data in an existing column of a first data type in the
4 database.

1 17) (Original) The method of claim 16 further comprising the step of maintaining
2 data coherency between the existing column and the new column.

1 18) (Original) The method of claim 12 further comprising the step of specifying a
2 preferred data type for at least one of a plurality of applications that access the database.

1 19) (Original) The method of claim 12 further comprising the step of gathering the
2 statistics.

1 20) (Original) A method for reading data from a database comprising the steps of:
2 (1) specifying a preferred data type for at least one of a plurality of applications
3 that access the database;
4 (2) detecting when one of the plurality of applications requests access to data in
5 the database;
6 (3) determining the preferred data type for the requesting application;
7 (4) determining if the data is stored in the database in the preferred data type for
8 the requesting application;
9 (5) if the data is stored in the database in the preferred data type for the requesting
10 application, returning the data to the requesting application;
11 (6) if the data is not stored in the database in the preferred data type for the
12 requesting application, performing the steps of:
13 (6A) converting the data to the preferred data type for the requesting
14 application; and
15 (6B) returning the converted data to the requesting application;
16 (7) reading statistics regarding the type of applications accessing data in the
17 database, the frequency with which the applications access the data, and the location of
18 the data being accessed by the applications; and
19 (8) dynamically changing a schema for the database to provide the preferred data
20 type when at least one of the plurality of applications requests access to data in the
21 database.

1 21) (Original) The method of claim 20 further comprising the step of gathering the
2 statistics.

1 22) (Original) The method of claim 20 wherein the step of dynamically changing the
2 schema for the database comprises the step of changing the data type of at least one
3 column in the database.

1 23) (Original) The method of claim 20 wherein the step of dynamically changing the
2 schema for the database comprises the step of adding a new column of a second data type
3 to the database that contains the same data in an existing column of a first data type in the
4 database.

1 24) (Original) A method for optimizing accesses to a database comprising the steps
2 of:

3 (1) reading statistics regarding the type of applications accessing data in the
4 database, the frequency with which the applications access the data, and the columns
5 being accessed by the applications;

6 (2) if the statistics indicate that a selected type of application has a number of
7 accesses to a selected column of a first data type in the database that exceeds a first
8 threshold level, performing the steps of:

9 (2A) if the statistics indicate that the selected type of application has a
10 number of accesses to the selected column that exceeds a second threshold level,
11 changing the data type of the selected column in the database;

12 (2B) if the statistics indicate that the selected type of application has a
13 number of accesses to the selected column that does not exceed a second
14 threshold level, adding a new column of a second data type to the database that
15 contains the same data in the selected column, the selected column and the new
16 column being defined as reflective columns because they contain the same data in
17 different data types.

1 25) (Original) The method of claim 24 wherein the first and second threshold levels
2 may be set by a human user via a user interface.

1 26) (Original) The method of claim 24 further comprising the step of maintaining
2 coherency of data in the selected column and the new column.

1 27) (Original) The method of claim 24 further comprising the step of gathering the
2 statistics.

1 28) (Original) The method of claim 24 further comprising the steps of:
2 (3) specifying a preferred data type for at least one of a plurality of applications
3 that access the database;
4 (4) detecting when one of the plurality of applications requests access to data in
5 the selected column;
6 (5) determining the preferred data type for the requesting application;
7 (6) determining if the data in the selected column is of the preferred data type for
8 the requesting application;
9 (7) if the data in the selected column is of the preferred data type for the
10 requesting application, returning the data in the selected column to the requesting
11 application;
12 (8) determining if the data in any column reflective of the selected column is of
13 the preferred data type for the requesting application;
14 (9) if the data in a reflective column is of the preferred data type for the requesting
15 application, returning the data from the reflective column to the requesting application;
16 (10) if the data in the selected column and in all reflective columns, if any, is not
17 of the preferred data type for the requesting application, performing the steps of:
18 (10A) converting the data to the preferred data type for the requesting
19 application; and
20 (10B) returning the converted data to the requesting application.

- 1 29) (Original) A program product comprising:
2 (A) a database optimizer that uses statistics regarding the type of applications
3 accessing data in a database, the frequency with which the applications access the data,
4 and the location of the data being accessed by the applications to make at least one
5 change to the database schema to optimize the performance of accessing data in the
6 database; and
7 (B) computer-readable signal bearing media bearing the database optimizer.
- 1 30) (Original) The program product of claim 29 wherein the computer-readable
2 signal bearing media comprises recordable media.
- 1 31) (Original) The program product of claim 29 wherein the computer-readable
2 signal bearing media comprises transmission media.
- 1 32) (Original) The program product of claim 29 wherein the database optimizer
2 makes the change to the database schema according to a set of rules that specify a
3 preferred data type for each type of application accessing data in the database.
- 1 33) (Original) The program product of claim 29 wherein the change to the database
2 schema comprises changing the data type of at least one column in the database.
- 1 34) (Original) The program product of claim 29 wherein the change to the database
2 schema comprises adding a new column of a second data type to the database that
3 contains the same data in an existing column of a first data type in the database.
- 1 35) (Original) The program product of claim 34 wherein the database optimizer
2 further comprises a data coherency mechanism for maintaining coherency between the
3 existing column and the new column.

1 36) (Original) The program product of claim 29 wherein the database optimizer
2 receives requests from at least one application to access data in the database, and returns
3 data from the database of a data type that is expected by the requesting application.

1 37) (Original) The program product of claim 29 wherein the database optimizer
2 further comprises a run-time statistics gathering mechanism to gather the statistics.

1 38) (Original) The program product of claim 29 wherein the database optimizer
2 operates according to customization settings set by a human user.

1 39) (Original) The program product of claim 29 wherein the database optimizer
2 further comprises a data type conversion mechanism that converts data in a first data type
3 retrieved from the database to a second data type that is preferred by an application
4 requesting the data.

1 40) (Original) A program product comprising:
2 (A) a database optimizer comprising:
3 a data access mechanism that uses statistics regarding the type of
4 applications accessing data in a database, the frequency with which the
5 applications access the data, and the location of the data being accessed by the
6 applications to make at least one change to the database schema to optimize the
7 performance of accessing data in the database;
8 customization settings that may be set by a human user to determine the
9 function of the database optimizer;
10 a data coherency mechanism that maintains coherency of reflective
11 columns in the database that are created by the data access mechanism and that
12 contain the same data in different data types; and
13 a data type conversion mechanism that converts data in a first data type
14 retrieved from the database to a second data type that is preferred by the
15 requesting application; and
16 (B) computer-readable signal bearing media bearing the database optimizer.

1 41) (Original) The program product of claim 40 wherein the computer-readable
2 signal bearing media comprises recordable media.

1 42) (Original) The program product of claim 40 wherein the computer-readable
2 signal bearing media comprises transmission media.

1 43) (Original) A program product comprising:
2 (A) a database optimizer comprising:
3 a mechanism that reads statistics regarding the type of applications
4 accessing data in the database, the frequency with which the applications access
5 the data, and the columns being accessed by the applications;
6 if the statistics indicate that a selected type of application has a number of
7 accesses to a selected column of a first data type in the database that exceeds a
8 first threshold level, the database optimizer determines whether the statistics
9 indicate that the selected type of application has a number of accesses to the
10 selected column that exceeds a second threshold level, and if so, the database
11 optimizer changes the data type of the selected column in the database;
12 if the statistics indicate that a selected type of application has a number of
13 accesses to a selected column of a first data type in the database that exceeds a
14 first threshold level, the data optimizer determines whether the statistics indicate
15 that the selected type of application has a number of accesses to the selected
16 column that exceeds a second threshold level, and if not, the data optimizer adds a
17 new column of a second data type to the database that contains the same data in
18 the selected column, the selected column and the new column being defined as
19 reflective columns because they contain the same data in different data types;
20 wherein the data optimizer detects when one of the plurality of
21 applications requests access to data in the selected column, determines the
22 preferred data type for the requesting application, determines if the data in the
23 selected column is of the preferred data type for the requesting application, and if
24 the data in the selected column is of the preferred data type for the requesting
25 application, returning the data in the selected column to the requesting
26 application;

(claim 43 continued)

27 if the data in any column reflective of the selected column is of the
28 preferred data type for the requesting application, the database optimizer returns
29 the data from the reflective column to the requesting application;
30 if the data in the selected column and in all reflective columns, if any, is
31 not of the preferred data type for the requesting application, the database
32 optimizer converts the data to the preferred data type for the requesting
33 application, and returns the converted data to the requesting application; and
34 (B) computer-readable signal bearing media bearing the database optimizer.

1 44) (Original) The program product of claim 43 wherein the computer-readable
2 signal bearing media comprises recordable media.

1 45) (Original) The program product of claim 43 wherein the computer-readable
2 signal bearing media comprises transmission media.

STATUS OF THE CLAIMS

Claims 1-45 were originally filed in this patent application. In the previous office action, claims 12 and 29 were objected to because of informalities. Claims 12, 14-16 and 18-19 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,418,451 to Maimone in view of U.S. Patent No. 6,529,901 to Chadhuri *et al.* (hereinafter “Chadhuri”). Claims 1-11 and 20-45 were allowed. Claims 13 and 17 were objected to as being dependent upon a rejected base claim, but would be allowable if properly rewritten in independent form. After a request for reconsideration from Applicants, the Examiner has maintained the rejection of claims 12, 14-16 and 18-19 and the objection of claims 13 and 17. In the present amendment, claim 12 has been amended and claim 13 has been cancelled. Claims 1-12 and 14- 45 are currently pending.